Pflichtmodul Informationssysteme (SS 2016)

Prof. Dr. Jens Teubner

Leitung der Übungen: Thomas Lindemann, Marcel Preuß

Übungsblatt Nr. 1

Ausgabe: 13.04.2016 Abgabe: 20.04.2016 – 12:00 Uhr

Aufgabe 1 (Die neun Codd'schen Regeln)

Informationssysteme müssen einer Menge von Eigenschaften genügen, um als Datenbankmanagementsysteme zu gelten. Eine solche Menge wurde von Edgar Codd definiert[1], die aus 9 Regeln besteht[2]:

1. Integration: Daten müssen in einer einheitlichen Struktur ohne Redundanz abgelegt werden.
2. Operationen: In einer Datenbank müssen Daten gespeichert, geändert, und gesucht werden können.
3. Katalog: Im Katalog werden Informationen abgelegt, die die Daten in der Datenbank beschreiben (z.B. welche Tabellen es gibt und welche Spalten sie haben).
4. Benutzersichten: Für unterschiedliche Anwendungen brauchen wir eine unterschiedliche Sicht auf den Datenbestand.
5. Integritätssicherung: Die Korrektheit des Datenbankinhalts muss gewährleistet werden, also kann man Regeln angeben, die jeder Datenbankeintrag erfüllen muss (z.B. Alter > 0).
6. Datenschutz: Nur authorisierte Benutzer/Programme dürfen auf die Datenbank zugreifen.
7. Transaktionen: Mehrere DB-Operationen müssen als eine funktionale Einheit ausgeführt werden, also entweder ganz (Commit) oder gar nicht (Transaktionsabbruch).
8. Synchronisation: Parallel ausgeführte Transaktionen müssen den gleichen Datenbankzustand hervorrufen wie irgendeine serielle Ausführung der Transaktionen.
9. Datensicherung: Das Datenbanksystem muss nach einem Systemfehler in der Lage sein, den letzten konsistenten Datenbankzustand wiederherzustellen.

[1] E. F. Codd. 1982. Relational database: a practical foundation for productivity. Commun. ACM 25, 2 (February 1982), 109-117. DOI=10.1145/358396.358400

[2] Saake, Gunter, Kai-Uwe Sattler, and Andreas Heuer. Datenbanken: Konzepte und Sprachen. Hüthig Jehle Rehm, 5. Auflage, Seiten 7–8, 2013.

Handelt es sich bei folgenden Systemen nach den 9 Codd'schen Regeln um Datenbankmanagementsysteme?

- MySQL, Excel, LSF, IBM DB2, iTunes, Microsoft SQLServer

Begründen Sie Ihre Antwort!

Aufgabe 2 (3-Ebenen-Schema Architektur und Datenunabhängigkeit)

Erläutern Sie das 3-Ebenen-Schema! Grenzen Sie dabei die Konzepte der einzelnen Ebenen von einander ab. Nennen Sie die Aspekte der Datenunabhängigkeit und erläutern Sie diese! Nehmen Sie dafür auch Bezug auf die Ausprägung der Datenunabhängigkeit im 3-Ebenen-Schema.

Aufgabe 3 (Zusatzaufgabe: Implementierung von Datenbankoperationen)

Auf der Vorlesungshomepage findet Ihr die Textdatei:

`presidents.txt`

Diese Datei enthält Informationen über einige Präsidenten der Vereinigten Staaten von Amerika. Die Datei ist entsprechend dem folgenden *Beispiel* strukturiert:

```
#presidents:[(Washington:Federalist:Virginia);...;(Obama:Democratic:Hawaii)]
#hobby:[(Jefferson:Fishing);(Jefferson:Riding);...;(Obama:Poker)]
```

Die Datei `presidents.txt` enthält also zunächst eine nichtleere Liste von Tripeln (von Strings). Dieser Liste ist das Schlüsselwort `#presidents:` vorangestellt, und sie beinhaltet Informationen über Präsidenten der USA. Die Liste wird durch eine eckige Klammer geöffnet und geschlossen. Die Tripel sind durch Semikola getrennt. Jedes Tripel wird durch eine runde Klammer geöffnet und geschlossen. Die Einträge der Tripels sind Strings und durch Doppelpunkte voneinander getrennt. Der erste String eines jeden Tripels bezeichne den Nachnamen des jeweiligen Präsidenten, der zweite String die Partei, welcher er angehört(e), und der dritte String seinen Geburtsstaat.

Der Liste der Präsidenten folgt in einer neuen Zeile eine nichtleere Liste von Paaren (von Strings). Dieser zweiten Liste ist das Schlüsselwort `#hobby:` vorangestellt, und sie beinhaltet Informationen über Hobbys der Präsidenten der USA. Die Liste wird ebenfalls durch eine eckige Klammer geöffnet und geschlossen. Die Paare sind durch Semikola getrennt. Jedes Paar wird durch eine runde Klammer geöffnet und geschlossen, und die Einträge sind Strings und durch Doppelpunkte voneinander getrennt. Der erste String eines jeden Paars bezeichne den Nachnamen des jeweiligen Präsidenten, der zweite String eines seiner Hobbys.

1. Schreibt in einer Programmiersprache Eurer Wahl ein Programm, welches die Informationen über die Präsidenten und ihre Hobbys aus der Datei ausliest und in einer geeigneten Datenstruktur abspeichert. (Es braucht dabei nicht dafür Sorge getragen zu werden, dass die Informationen geändert oder sessionsübergreifend gespeichert werden können.)

2. Erstellt dann Methoden zur Darstellung der folgenden Informationen in der Bildschirmausgabe:

   (a) die Namen aller Präsidenten, welche der republikanischen Partei (`Republican`) angehören oder angehörten.

   (b) die Namen aller Präsidenten, welche der republikanischen Partei (`Republican`) angehören oder angehörten, in alphabetischer Reihenfolge.

   (c) die Hobbys aller Präsidenten, welche *nicht* der demokratischen Partei (`Democratic`) angehören. Diese Informationen sollen in Form von Paaren der Form `(party:hobby)` ausgegeben werden, wobei `hobby` ein Hobby eines Präsidenten ist, welcher der Partei `party` angehört.

   (d) für jeden Präsidenten die Anzahl seiner Hobbys. Es sollen Paare der Form `(name:number_of_hobbies)` ausgegeben werden, wobei `number_of_hobbies` die Anzahl der Hobbys des Präsidenten angibt.

3. Überlegt anhand der Implementierung, ob und wie sich dieser Ansatz, Informationen und Daten abzuspeichern und darauf zu operieren, verallgemeinern lässt. Überlegt insbesondere, was passiert, wenn weitere Listen mit Informationen über die Präsidenten hinzugefügt würden (zum Beispiel über die Ehen der Präsidenten) oder wenn die Struktur der Tupel innerhalb einer Liste geändert würde (wenn zum Beispiel jedem Präsidententupel noch sein Todesalter hinzugefügt würde). Diskutiert die Probleme in der Übung!